An Evolutionary Coupled Neural Oscillators with Application to Pattern Recognition

A. Tobal[*] and K. Mahar[**]

College of Computer and Information Sciences, King Saud University,
Riyadh, Saudi Arabia
E-mail: atobal@ccis.ksu.edu.sa
** *Computer Engineering Dept., College of Engineering and Technology,*
Arab Academy for Science and Technology, Alexandria, P.O.Box 1029, Egypt,
E-mail: khmahar@aast.edu

Abstract. Cyclical activities are basic characteristics of all living organisms. Neurobiologists have discovered that a single neuron often possesses membrane properties that are responsible for the generation of oscillations. When coupled with other neurons, oscillations with varying properties depending on the type of interconnection can be generated. Using synchronization and temporal correlation of these oscillations can carry out the tasks of pattern recognition of different objects. The speed of recognition depends on the speed of synchronization. In this paper, we propose evolutionary coupled neural oscillators to minimize the time of synchronization through the optimization of the neuron parameters by means of a genetic algorithm. The genetic algorithm, with its global search capability, finds the optimum neuron parameters through a fitness measure that reflects the correlation strength between oscillators. Thus avoiding the trial-and-error process of estimating the neuron parameters. The superiority of the method is demonstrated through an application of character recognition process.

Introduction

Artificial Neural network (ANN) models can be classified, according to their computational units, into three different generations. The first generation is based on McCulloch-Pitts neurons as computational units [1]. These are also referred to as perceptrons or threshold-gates. They give rise to a variety of neural network models such as Hopfield nets and Boltzman machines. A characteristic feature of these models is that they can only give digital output, and every Boolean function can be computed by some multi-layer perceptron with a single hidden layer.

The second generation is based on computational units that apply to weighted-sum of inputs of an activation function with a continuous set of possible output values. Typical examples of networks from this generation are feed-forward and recurrent sigmoidal nets. It has been shown that these nets can compute certain Boolean functions with fewer gates than neural nets from the first generation. In addition they are able to compute functions with analog input and output. Another characteristic feature of these second-generation models is that they support learning algorithms such as back-propagation. For a biological representation of neural nets, from the second-generation, the output of a sigmoidal unit represents the firing rates of a biological neuron.

Recently, experimental evidence has accumulated and indicated that many biological neural systems use timing of single action potentials (or spikes) to encode information. These experimental results from neurobiology have lead to the investigation of a third generation of neural network models, which employ spiking neurons or "integrate and fire neurons" as computational units [2]. In particular these models describe, in a much better way, the actual output of a biological neuron, and hence they allow for using time as a resource for computation and communication. The timing of individual computation steps play a key-role in computations in networks of spiking neurons whereas the timing of computation steps is usually trivialized in the preceding two generations; either through an assumed synchronization, or through an assumed stochastic asynchronicity.

The above discussion has revealed that the third generation of NN models may be used to build a coupled neural oscillator model for pattern recognition applications. The idea is based on the fundamental aspect of perception, where binding spatially separate sensory features, is essential for object recognition, segmentation of different objects, and figure ground segregation [3;4]. We have found that locally coupled neural oscillators can indeed yield global synchrony [14]. It has been shown that with the used model, synchronization is stable and phase locking can be controlled through the model parameters. This leads to the question whether the system parameters tuning can improve the network performance?

This paper deals with the system parameters calculation to improve the network performance. The aim is to optimize the parameters of the neuron model to speed up the synchronization of the network neurons, thus speeding up the recognition process. The optimization is carried out by means of genetic algorithms or neuro-evolution. Neuro-evolution is the paradigm of evolving artificial neural network through genetic algorithm. It is effective for finding optimum neural network parameters in tasks where the only information from the environment comes from an evaluation function, or where no training examples can be provided. Recent work has shown that evolved neural networks are effective in several areas such as dynamic control tasks [5], and data classification [6].

Until Now, the incorporation of genetic algorithms and coupled oscillators neural networks may have not been addressed. But, on the other hand, various schemes for augmenting genetic algorithms and second-generation neural nets have been proposed in recent years. In this regard, the simplest scheme uses the genetic algorithms as a stand-alone learning algorithm for ANN [7]. Another scheme where a network is represented as a genotype that has six kinds of genes was proposed in [8]. The genes are a learning rate, a slant of sigmoid function, a coefficient of momentum term, an initializing weights range, the number of layers and the unit numbers of each layer. Genetic operators affect populations of these genotypes to produce adaptive networks with higher fitness values. In [9], Larranaga et al. presented an approach to structure learning in the field of Bayesian networks. They tackled the problem of the search for the best Bayesian network structure, given a database of cases, using a genetic algorithm philosophy for searching among alternative structures. Another idea of augmenting genetic algorithm with ANN was presented in [10] where a genetic algorithm searches among candidate solutions of the problem, while the neural network provides the objective function value of each candidate solution. This method is suitable for problems where the evaluation of the object function is computationally time- consuming and may seem ill fitted.

In most approaches of neuro-evolution, each individual represents a complete neural network that is evaluated independently of other networks in the population. Thus, each neuron is evaluated only with the other neurons encoded on the same chromosome. Consequently, a very good neuron may exist on a chromosome but is subsequently lost because the other neurons on the chromosome are poor [11]. Fortunately, this problem is not encountered in our coupled oscillator neurons, where we consider the network is built from interconnecting similar neurons or individuals. Hence the evolution is at the neuron level which accurately evaluates the fundamental building block of the network by evaluating how it performs when combined with each other.

This paper is organized as follows: the next section motivates and describes the used spiking model and its computer simulation. Section III describes the encoding scheme of the neuron parameters and explains the mechanism of the evolution process. The model simulation results are the subject of section IV, while section V contains the conclusions and directions for future work.

Model Description and Simulation

While many crucial properties of real neurons remain unknown, biophysical neural models at least attempt to incorporate some known properties of the neural tissue. Like real neurons, these models produce spikes rather than continuous-valued outputs [12]. There are a lot of such models, but in this work, Ellias-Grossberg model [13] was chosen because of its simplicity, speed of synchronization compared to the other biological neural network models, ease of control due to its parameters and stability of synchronization [14]. As the building block, the model of a single oscillator is defined,

in the simplest form, as a feedback loop between an excitatory unit and an inhibitory unit [15]-[16]. The model can be described by the following two equations:

$$\frac{dx}{dt} = -Ax + (B-x)\{C[x-\Gamma]^+ + I\} - Dx[y-\Gamma]^+... \tag{1}$$

$$\frac{dy}{dt} = E(x-y) \tag{2}$$

Where $[s]+ = \max (s, 0)$. The variable x represents the potential of an excitatory cell governed by a nonlinear shunting equation, and y represents the potential of an inhibitory cell governed by a linear equation. The parameters are $A=1$, $B=1$, $C=20$, $D=33.3$, $\Gamma=0.5$ as reported in many papers [13], [14] and I ranges from 0.2 to 1.0 in different trials. E in (2) governs the relative time scales of x and y and represents the relative rate at which the inhibitory interneuron tracks the firing rate of the excitatory cell. When E is small and thus (2) is in its relaxation regime, the excitatory cell exhibits a spike like waveform, and when E is near unity the excitatory cell exhibits sinusoidal behavior. Figure 1 shows the single oscillator model state representation and its simulation results in the relaxation regime for the neuron output activities, and its phase plane analysis.
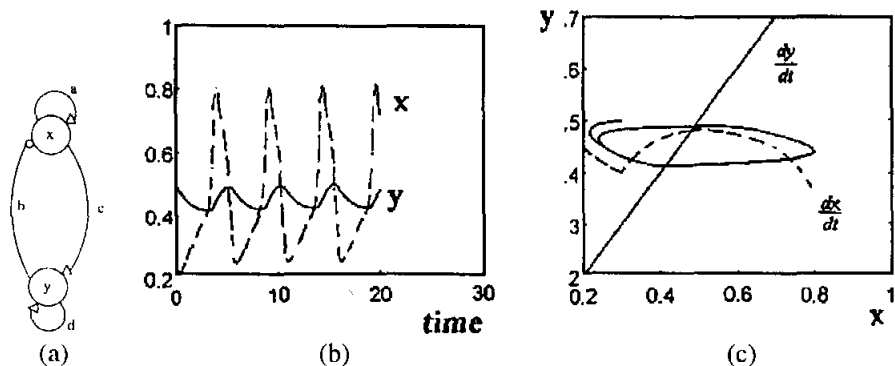

Fig. 1. Relaxation oscillation obtained for Ellias-Grossberg model. (a) The model state representation, (b) Waveforms for inhibitory cell (solid curve) and excitatory cell (dashed curve), (c) Phase portrait, limit cycle and Nullclines [14].

Using the Ellias-Grossberg Model, we implemented a computer simulation for two-dimensional layer of 10x22 oscillators with the connection scheme shown in Fig. 2-c. In this study the network connections were implemented for 4 nearest neighbors in the excitatory layer while the inhibitory cell is coupled only to its coupled Excitatory cell.

The ith oscillator was governed by (1) and (2), where x and y were replaced by x_i and y_i, respectively. Nearest neighbor coupling was implemented by adding the term

$$\xi \alpha(B - x_i)\{ \sum_{k \in Ni}[x_k - \Gamma]^+ \} \tag{3}$$

where (3) governs the excitatory coupling term that is added to equation (1) in population. (2) governs the inhibitory coupling term if the inhibitory layer connections are not taken into consideration. α is the excitatory coupling strength, N_i is the set of the adjacent oscillators that connect to oscillator i and ζ is a scaling factor calculated according to the number of connected neighbors [14].
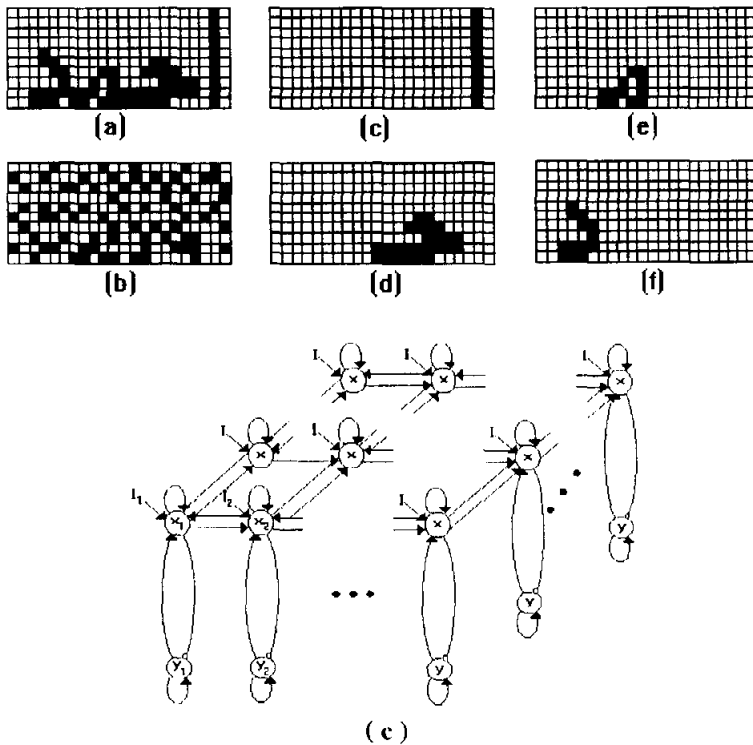

Fig. 2. Object "A " is selected as an example of the scene to the network to show the effect of changing the model parameters. (a) The output activities are randomly initialized. (b) The output activities at the synchronization time. (c) The network interconnections.

For nearly all sets of initial phase conditions, the relaxation arrays exhibited rapid synchronization at some moderate coupling strength, while at moderate coupling strength

the sinusoid arrays exhibited slow synchronization or no synchronization. We found that changing the model parameters affects the time of the network synchronization that leads to the question whether the system parameters tuning can improve the network · performance? The next section deals with the system parameters calculation to improve the network performance.

The Evolution Process

Evolutionary algorithms are probabilistic search algorithms that simulate natural evolution. Genetic algorithms (GAs) are one of these types of algorithms [17]. They are based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures. In GAs the search space of the problem is represented as a collection of individuals. The individuals are represented by character strings, which are often referred to as chromosomes. The purpose of the use of a GA is to find the individual from the search space with the best genetic material. The quality of an individual is measured with an objective function. The part of the search space, which is to be examined, is called the population.

Roughly, a GA works as follows: First, the initial population is chosen, and the quality of each individual is determined. Next, in every iteration, parents are selected from the population. These parents produce children, which are added to the population. For all newly created individuals a probability near zero exists that they mutate. After that, some individuals are removed from the population according to a selection criterion in order to reduce the population to its initial size. Each iteration of the algorithm is referred to as a generation.

The operators that define the child production process and the mutation process are called the crossover operator and the mutation operator, respectively. Mutation is needed to explore new states and helps the algorithm to avoid local optima. Crossover should increase the average quality of the population. By choosing adequate crossover and mutation operators as well as an appropriate reduction mechanism, the probability that the GA results in a near-optimal solution in a reasonable number of iterations increases.

In our evolutionary, or genetic, approach, each individual, or chromosome, in the neuron population specifies a set of connections (excitatory and inhibitory) to be made with their neighbors. Each connection is encoded as a floating-point number and is regarded as a gene of the chromosome. From Equation (1) and Fig.1 (a) we find that the excitatory and inhibitory connections are A, B, C, D, β, α and Γ. These parameters need to be evolved to enhance the performance of the oscillator. Consequently they have to be regarded as the elements of the chromosome.

The neuron evolution uses the basic strategy of GAs for evaluating and recombining the fittest individual neurons. As stated above, the strategy is repeated over two phases: an evaluation phase and a reproduction phase. During the evaluation phase, neuron parameters are evaluated based on the performance of the network in which they participate. The performance is measured as the speed of synchronization and correlation between the coupled neurons that is calculated according to Equation 4. The fitness measure that we employ is given by Equation 5.

$$
\left.\begin{aligned}
&\mathrm{corr}(t) = \sum_{i=1\to n} \varphi(x(i,t)) \\
&\varphi(x(i,t)) = 1 \quad\text{if}\quad x(i,t) \geq \theta \quad\text{and}\quad [\sum_{k\in N_i, k\neq i} x(k,t)] \geq \theta
\end{aligned}\right\} \tag{4}
$$

$$
\varphi(x(i,t)) = 0 \qquad \text{Otherwise}
$$

$$
f = \sum_{t=0\to t_f} \max(x(t) - \Gamma - 0.2, 0) * (\mathrm{corr}(t) - \mathrm{corrd}(t))^2 \tag{5}
$$

where f represents fitness function, $x(t)$ is the output activities at time t, corr(t) is the correlation function output at time t and corrd(t) is the desired correlation function output at time t. The function ‚ˆˆ L'Q GL F DW he neuron is correlated at least to an adjacent neuron or not at all. The function takes only the binary values 1 and 0, respectively.

In the reproduction phase, genetic operators which are selected by rank, one-point crossovers, and mutation, are used to obtain new neurons. Every two neurons in the top 50% of the population (according to the fitness rank), are selected for mating. Each mating operation creates two offspring through arithmetic crossover. This process produces two complementary linear combinations of the parents X and Y as:

$$
\left.\begin{aligned}
&\overline{X} = rX + (1-r)Y, \text{and} \\
&\overline{Y} = (1-r)X + rY
\end{aligned}\right\} \tag{6}
$$

where r is a uniform random number between 0 and 1.

The new offspring replace the bottom 50% worst performing neurons in the population. Finally, mutation at a rate of 1% was implemented by randomly selecting one variable and setting it equals to a uniform random number $u(a_i, b_i)$, where a_i and b_i are the lower and upper bound of the variable respectively. The fittest individual was not mutated, but copied to the next generation to ensure the survival of the best solution till the end of the process.

Experimental Results

The test pattern used, as an example in this paper, is an image of the "A" pattern, mapped to a 10X22 network. If a square is entirely covered by the input, the corresponding oscillator receives an external input; otherwise, the oscillator receives no external input. The oscillators receiving no external input are shown in the bottom of the display in Fig. 3 for the test pattern. The activities of the oscillators simulated by the object are combined into a single trace in the Figure. Thus if they are synchronized, they appear like a single oscillator.

Several Iterations were done with different parameter values for the used model with the test pattern. The same parameter values specified before were used with $\alpha=0.2$, and ζ was chosen according to the number of connected neighbors which is 0.25 in our case. The applied input I used was 0.8 for the nodes receiving external input and 0.0 for the rest. Random values ranging from 0.0 to 0.5 are used as the initial values of the inhibitory and excitatory action potential voltages. Changing the initial values does not affect the time of synchronization but affects the first part of the resulting traces.

By applying the GA, the neurons were evolved for 50 generations. After evolution, the best neuron was used to build the network. The results are shown in Fig. 3 for the reported and the calculated parameters by means of the used genetic algorithm. With regard to the synchronization speed, it is clear that the evolved parameters are effective than the reported values. By studying and comparing the two parts of Fig. 3, we found that the GA with its ability of finding global optima enhances the speed of synchronization, at least to a value twice that of the reported values.

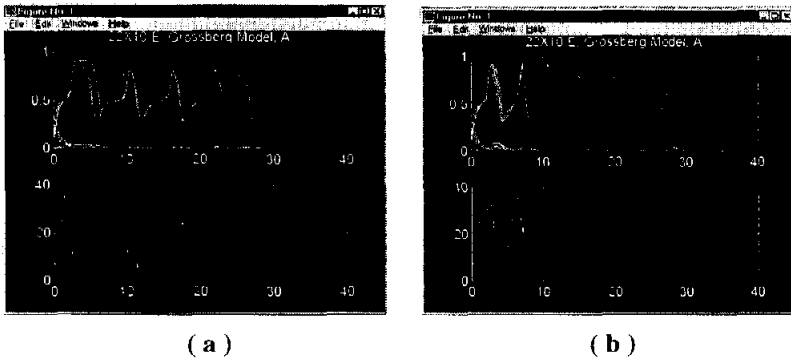(a)   (b)

Fig. 3. The upper traces show the combined temporal activities of the oscillator blocks representing the character 'A'. Each Excitatory cell is connected to its 4 immediate neighbors except for the boundaries, no connections in the inhibitory layer and no wrap-around are used. The Lower trace is the temporal Correlation output. (a) The best try and error parameters used A=B=1, C=20, D=33.3, E=0.2, ς=0.25, α= 0.2, Γ= 0.45 and I = 0.8 for the nodes receiving external input and equals to zero for the rest.

(b) The resultant GA parameters used are A=B=1, C=21.25 , D=26.76, E=0.36, ς=0.25, α=0.41, Γ= 0.49.

An expanded network based on the same parameter values is tested for some connected Arabic words to test the ability of the network to concurrently segment and recognize in short time compared to the other networks [4], [15], [18]. The same improvement we have showed in this paper for the given example is achieved for all characters.

Conclusion

In this paper, we presented evolutionary coupled neural oscillators and demonstrated their capabilities in segmentation and recognition of image patterns. Like real neurons, neural oscillators produce spikes rather than continuous-valued outputs. There are a lot of such models, but in this paper, the Ellias-Grossberg model was chosen as the building block of the network. This is because of its simplicity, speed of synchronization compared to the other biological neural network models, ease of control due to its parameters, insensitivity to a wide range of its parameters and stability of synchronization. The intent of the evolution process is the minimization of the time of synchronization through optimizing the neuron parameters. The neuron evolution uses the basic strategy of GAs for evaluating and recombining the fittest individual neurons. The strategy is repeated over two phases: an evaluation phase and a reproduction phase. During the evaluation phase, neuron parameters are evaluated based on the performance of the network in which they participate. The Obtained results indicate that GA, with its global search capability, finds the optimum neuron parameters. These evolved parameters enhanced the performance of the network at least two times better compared to the previous reported parameters. Also, the use of GA eliminates the trial-and-error process of estimating the neuron parameters. In future, we will try to use genetic algorithms to find a compromise between increasing the network structure and the needed accuracy of segmentation and recognition.

References

[1]   Softky, W. and Koch, C. "Single-cell Models." The Handbook of Brain Theory and Neural Networks, The MIT Press, Cambridge, Massachusetts, London: England, (1995), 879-884.

[2]   FitzHugh, R.A. "Impulses and Physiological States in Theoretical Models of Nerve Membrane." *Biophys. J.,* 1 (1996), 445-466.

[3]   Wang, D.L. and Terman, D. "Image Segmentation Based on Oscillatory Correlation." *Neural Comp.,* 9 (1997), 805-836.

[4]   Singer, D. and Gray, C.M. "Visual Feature Integration and Temporal Correlation Hypothesis." *Ann. Rev. Neurosci.,* 18 (1995), 55-586.

[5]   Moriarty, D.E. and Miikkulainen, R. "Efficient Reinforcement Learning Through Symbiotic Evolution." *Machine Learning,* 22 (1996), 11-32.

[6]   Ignizio, J.P. and Soltys, J.R. "Simultaneous Design and Training of Ontogenic Neural Network Classifier." *Computers Ops Res.,* 23, No. 6 (1996), 535-546.

[7]   Janson, D.J. and Frenzel, J.F. "Application of Genetic Algorithms to the Training of Higher Order Neural Networks." *Journal of Systems Engineering,* 2, Iss. 4 (1992), 272-276.

[8]   Takahashi, H., Agui, T. and Nagahashi, H.D. "Designing Adaptive Neural Network Architectures and their Learning Parameters Using Genetic Algorithms." *Proceedings of the SPIE - The International Society for Optical engineering,* vol. 1966, pp. 208-215, 1993.

[9] Larranaga, P., Poza, M., Yurramendi, Y., Murga, R.H. and Kuijpers, C.M.H. "Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters." *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18 No. 9 (Sep. 1996), 912-925.

[10] Coit, D.W. and Smith, A. E. "Solving the Redundancy Allocation Problem Using a Combined Neural Network/Genetic Algorithm Approach." *Computer Ops Res.*, 23 No. 6 (1996), 515-526.

[11] Moriarty-2m D.E. and Miikkulainen, R. "Hierarchical Evolution of Neural Networks." *Proceedings of the IEEE Conference on Evolutionary Computation* (ICEC-98, Anchorage, AK), Piscataway, NJ:IEEE, 1998.

[12] Freeman, W.J. "Tutorial on Neurobiology from Single Neuron to Brain Chaos." *International Journal of Biforcation and Chaos*, 2, No. 3 (September, 1992), 451-482.

[13] Ellias, S.A. and Grossberg., S. "Pattern Formation, Contrast Control, and Oscillations in the Short term Memory of Shunting On-center Off-surround Networks." *Biol. Cybern.* 20 (1975), 69-98.

[14] Ahson, S.I. and Mahmoud A.M. "Pattern Segmentation Using Coupled Neural Oscillators." WCNA, Athens: Greece, July 8-11, 1996.

[15] Wang, D. "Emergent Synchrony in Locally Coupled Neural Oscillators." *IEEE Trans. Neural Networks*, 6, No. 4 (July, 1995), 941-948.

[16] Somers, D. and Kopell, N. "Rapid Synchronization Through Fast Threshold Modulation ." *Biol. Cybern.*, 68 (1993), 393-407.

[17] Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1988,

[18] Compbell, S. and Wang, D. "Synchronization and Desynchronization in a Network of Locally Coupled Wilson-Cowan Oscillators." *IEEE Trans. Neural Networks*, 7, No. 3 (May 1996), 541-553.

مذبذبات عصبية متعانقة تطوّرية و تطبيقها في التعرف على الأنماط

د. أحمد محمود توبال* و د. خالد محار**

*كلية علوم الحاسب والمعلومات، جامعة الملك سعود ص. ب ٥١١٧٨، الرياض ١١٥٤٣،
السعودية، بريد إلكتروني: *Error! Bookmark not defined.*
**كلية الهندسة والتكنولوجيا، الأكاديمية العربية للعلوم والتكنولوجيا، إسكندرية، ص.ب. ١٠٢٩، مصر،
بريد إلكتروني: *Error! Bookmark not defined.*

ملخص البحث. تعتبر النشاطات الدورية أساسا لاستمرارية حياة الكائنات الحية. واكتشف علماء الأحياء أن الخلية العصبية المنفردة تنتج ذبذبات ذات تردد وشكل مميز يعتمد على خواص الغشاء المحتوي على الخلية. كذلك عند ربط أكثر من خلية عصبية مع بعضها البعض ، فإنه ينتج عنها ذبذبات تعتمد على نوع الوصلات فيما بينها. وبدراسة تزامن هذه الذبذبات والعلاقة المؤقتة فيما بينها وجد أنها تحقق إمكانية التعرف على الأنماط. وتعتمد سرعة التعرف على الأنماط على سرعة التزامن بين خلايا المجموعة المرتبطة. ومن ثم فإننا نعرض في هذا البحث مذبذبات عصبية متعانقة تطوّرية يمكنها زيادة سرعة تزامنها. ويتم ذلك عن طريق البحث عن معاملات مثالية للخلية العصبية باستخدام خوار زم تطوّري يعتمد على قياس مدى قوة العلاقة بين المذبذبات. وبهذه الطريقة يتم تفادي استخدام عملية التجربة والخطأ والتي تستخدم في استنباط معاملات الخلية العصبية. وتجلى بوضوح تفوق هذه الطريقة على غيرها من خلال تطبيقها في عملية التعرف على الأنماط.